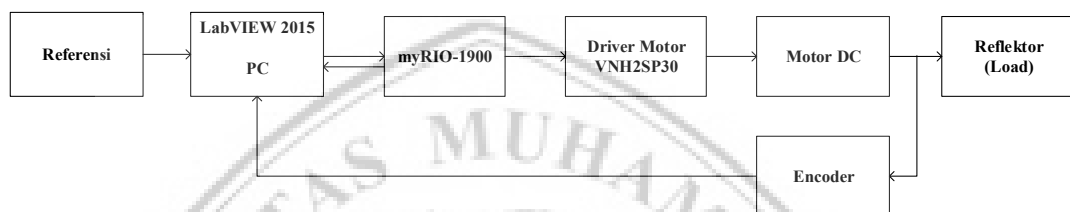


BAB III

PERANCANGAN DAN PEMBUATAN SISTEM

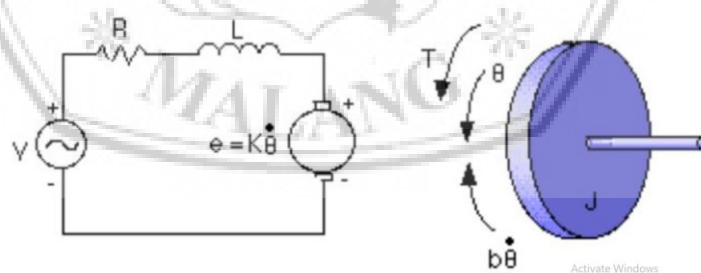
Dalam bab ini akan menjelaskan perancangan dan pembuatan sistem pada kontrol secara struktur global maupun desain kontroler yang digunakan untuk mendukung keseluruhan sistem yang akan diuji. Beberapa komponen pendukung pada sistem ini ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram Blok Perencanaan Sistem

3.1 Pemodelan Motor DC

Sebelum masuk ke pembuatan kontrol LQR penelitian ini terlebih dahulu merancang kontrol sistem dengan menggunakan jenis struktur model *state space* (ruang-keadaan) dimana pada tahap awal pembuatannya dengan memodelkan sistem dari motor DC sebagai aktuator. Gambar 3.2 model skematik dari motor DC.



Gambar 3.2 Diagram Skematik MotorDC

Untuk motor DC pada gambar diatas didapatkan nilai parameter yang diperoleh dari spesifikasi pada motor DC yang sudah diketahui dan didefinisikan dalam tabel 3.1

Tabel 3.1 Parameter dari Model Motor DC Faulhaber

| Faulhaber 2342L012CR | | |
|----------------------|--|--------|
| Parameter | Definisi | Nilai |
| V_a | Motor Voltage[V] | 12 |
| R_a | Armature Resistance[Ω] | 1,9 |
| Eff_m | Efficiency max.[%] | 80 |
| P_m | Motor power[W] | 17 |
| n_0 | No-load speed[rad] | 8100 |
| L_a / M_H | Motor Inductance[H] | 0,6 |
| B_m / M_R | Friction torque[Nms/rad] | 0.01 |
| a_m | Motor and load pole | 1.7 |
| K_n | Speed constant[rad/V] | 713 |
| K_e / K_B | Back-EMF constant[V/s/rad] | 0.014 |
| K_m / K_T | Torque constant[Nm/A] | 0.0134 |
| K_l | Current constant[A/Nm] | 0.07 |
| B_a | Motor damping constant[Nms/rad] | 0.01 |
| J_a | Motor inertia[Kgm ²] | 0.05 |
| B_l | Load damping constant[Nms/rad] | 1 |
| J_l | Load inertia constant[Kgm ²] | 1 |
| J_m | Equivalent moment inertia[Kgm ²] | 0.06 |
| K_g | Gear ratio | 66 |

Dari parameter-parameter yang sudah ditentukan persamaan sistem dengan torsi motor T , berhubungan dengan arus armatur i , dengan faktor konstan K_T . Dimana Back-EMF e , berhubungan dengan kecepatan rotasi dengan persamaan berikut:

$$T = K_T i \quad (3.1)$$

$$e = K_e \theta \quad (3.2)$$

Dalam satuan SI yang akan digunakan , $K_T = Ke$, dari gambar 3.2 maka didapatkan persamaan berikut berdasarkan hukum Newton dan Kirchhoff yang digabungkan pada persamaan 3.3 dan 3.4:

$$J_a \frac{d^2\theta}{dt^2} + B_a \frac{d\theta}{dt} = K_T I_a(t) \quad (3.3)$$

$$L_a \frac{dI_a(t)}{dt} + R_a I_a(t) = V_a(t) - K_T \frac{d\theta}{dt} \quad (3.4)$$

Menerapkan *transformasi laplace* ke persamaan 3.3 dan 3.4 dengan kondisi nilai awalan 0 atau dinyatakan dalam bentuk s :

$$J_a s^2 \theta(s) + B_a s \theta(s) = K_T I_a(s) \quad (3.5)$$

$$L_a s I_a(s) + R_a I_a(s) = V_a(s) - K_T s \theta(s) \quad (3.6)$$

Dengan menghilangkan variabel $I_a(s)$ membuat subjek pada persamaan 3.6 dan mensubstitusikan kedalam persamaan 3.5, dan menghasilkan persamaan 3.7:

$$J_a s^2 \theta(s) + B_a s \theta(s) = K_T \frac{V_a(s) - K_T s \theta(s)}{R_a + L_a s} \quad (3.7)$$

Dari persamaan 3.7, fungsi transfer dari tegangan *input* $V_a(s)$ ke *output* sudut θ secara langsung mengikuti persamaan 3.8:

$$G_a(s) \frac{\theta(s)}{V_a(s)} = \frac{K_T}{s[(R_a + L_a s)(J_a s + B_a) + K_T^2]} \quad (3.8)$$

Bentuk *state space* dari persamaan diatas dinyatakan dengan memilih kecepatan putar, posisi keluaran dan arus sebagai variabel keadaan dan tegangan sebagai masukan. *Outputnya* dipilih menjadi kecepatan putar dan posisi. Sistem *Linear-Time-Invariant* (LTI) diberikan umumnya oleh persamaan 3.9 dan 3.10:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \quad (3.9)$$

$$\frac{d}{dt} \begin{bmatrix} \theta \\ \dot{\theta} \\ I_a \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{B_a}{J_a} & \frac{K_T}{J_a} \\ 0 & -\frac{K_T}{L_a} & \frac{R_a}{L_a} \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ I_a \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} V \quad (3.10)$$

$$y = [1 \quad 0 \quad 0] \begin{bmatrix} \theta \\ \dot{\theta} \\ I_a \end{bmatrix}$$

Dengan membuat persamaan model *state space* di program LabVIEW maka akan lebih mudah dalam mendapatkan hasil karena pada model *state space* tidak memungkinkan untuk menghitung dengan tangan. Kemudian untuk mendapatkan hasil yang sama dengan pendekatan hibrida, maka digunakan MathScript Node dengan memasukkan parameter motor DC pada tabel 3.1:

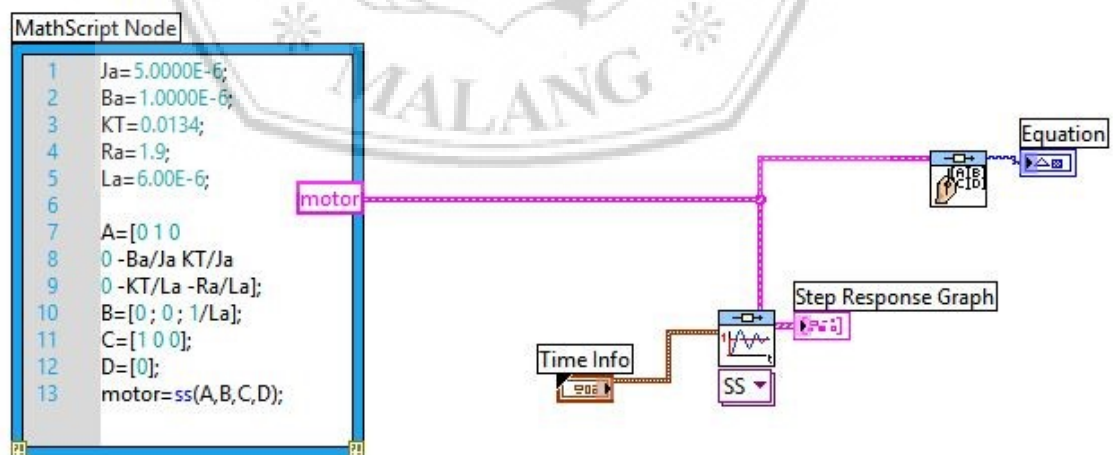
```

Ja = 5,0000E-6;
Ba = 1,0000E-6;
KT = 0,0134;
Ra = 1,9;
La = 6.00E-6;

A = [0 1 0
      0 -Ba / JaKT / Ja
      0 -KT / La -Ra / La];
B = [0; 0; 1 / La];
C = [1 0 0];
D = [0];
motor = ss (A, B, C, D);

```

Hasil pada program LabVIEW dengan MathScript Node dengan parameter menggunakan model state space:



Gambar 3.3 MathScript Node dengan Model State Space

State Space Equations

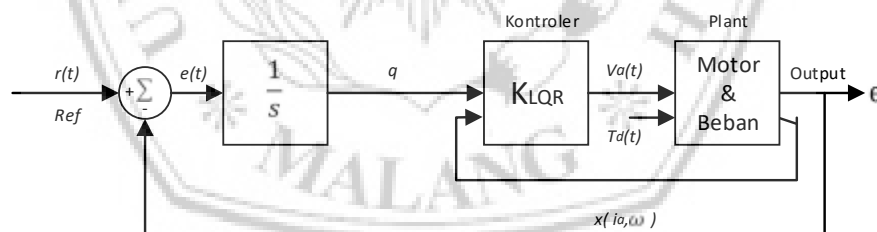
$$\frac{dx}{dt} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -0,2 & 2680 \\ 0 & -2233,33 & -316667 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ 166667 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \end{bmatrix} u(t)$$

Gambar 3.4 Persamaan State Space di LabVIEW

3.2 Perancangan Kontrol LQR

Adapun blok diagram pada kontrol LQR yang telah dirancang seperti pada Gambar 3.5. Langkah pertama adalah memilih matriks-matriks dalam persamaan ini sebagai Aa , Ba , Ca , dan Da , dimana matriks-matriks ini nanti akan mengacu pada vektor status dari sistem yang diperbesar sebagai Xa . Referensi, r , tidak mempengaruhi state (kecuali state integrator) atau keluaran plant, karena tidak ada jalur dari referensi ke input plant, u , tanpa menerapkan matriks umpan balik. Selanjutnya, vektor gain umpan balik konstanta K dihitung menggunakan program LabVIEW dengan respon langkah sistem *close-loop* yang ditemukan melalui simulasi.



Gambar 3.5 Kontrol Sistem LQR

Dalam hal ini diperlukan sebuah masukan integrator ekstra secara seri dengan plant agar dapat menghilangkan *steady-state error* ke sebuah inputan. Fungsi dari integrator itu sendiri adalah untuk meregulasi gangguan yang diberikan sehingga gangguan itu tidak akan mempengaruhi sistem pada saat *steady state*. Pemodelan integrator dengan menambahkan persamaan state dengan ekstra state yang merupakan integral dari output dimana turunan dari persamaan tersebut adalah

$\int \theta = \theta$. Sehingga muncul dalam persamaan state yang baru ditambahkan dengan output sistem tetap sama.

$$\frac{d}{dt} \begin{bmatrix} \int \theta \\ \theta \\ \dot{\theta} \\ I_a \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{B_a}{I_a} & \frac{K_T}{I_a} \\ 0 & 0 & -\frac{K_T}{L_a} & \frac{R_a}{L_a} \end{bmatrix} \begin{bmatrix} \int \theta \\ \theta \\ \dot{\theta} \\ I_a \end{bmatrix} + \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \end{bmatrix} r \quad (3.11)$$

$$y = [0 \quad 1 \quad 0 \quad 0] \begin{bmatrix} \int \theta \\ \theta \\ \dot{\theta} \\ I_a \end{bmatrix}$$

Dari data yang diperoleh diatas dengan menggunakan pendekatan integrator *state-space* untuk menganalisa dan mengendalikan sistem LTI yang model umum diwakili dalam persamaan 3.11 maka untuk memulai desain LQR dilakukan penyusunan sebagai berikut:

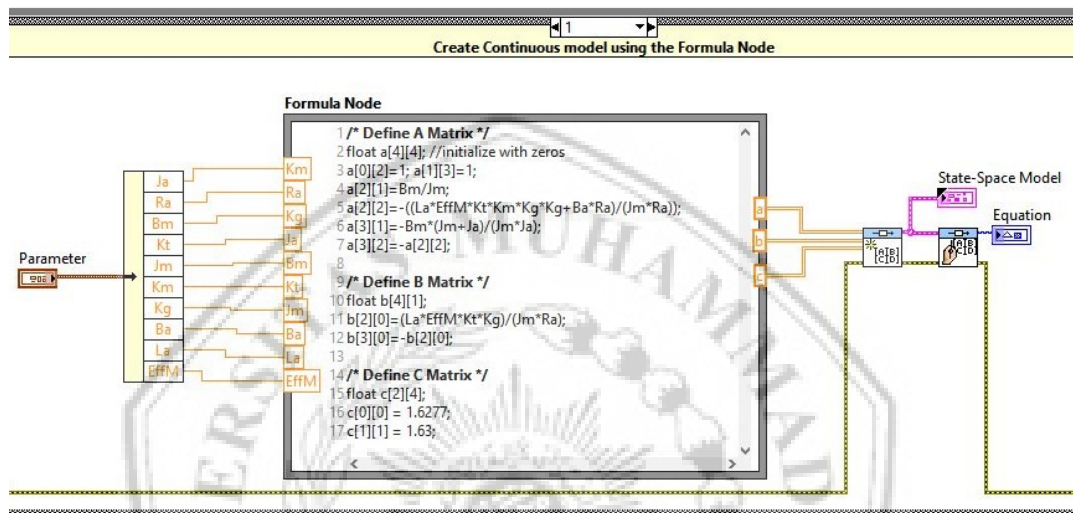
1. Diketahui model *plant* yang sudah didefinisikan pada tabel 3.1 dan persamaan 3.11.
2. Semua keadaan dapat diukur secara langsung dengan parameter yang sudah didapatkan.

Jika susunan pertama tidak benar (model memiliki ketidakpastian), maka digunakan kontrol robust lebih diutamakan. Jika susunan kedua tidak benar (hanya output yang terukur), maka gangguan dalam hal ini sistemnya harus *observable*. Metode LQR digunakan untuk mencapai kinerja sistem yang dapat diterima dengan meminimalkan performa indeks (*cost*) nilai J yang didefinisikan untuk *stedy state case* dengan persamaan sebelumnya, maka dibuatlah formulasi model *state space* dari parameter yang sudah didapatkan dimana nilai A dan B diperoleh dari definisi variabel:

$$\begin{bmatrix} \int \theta \\ \theta \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{Bm}{Jm} & \frac{-Effm La Kt Km Kg^2 + Ba Ra}{Jm Ra} & 0 \\ 0 & \frac{-Bm(Jm+Ja)}{Jm Ja} & \frac{Effm La Kt Km Kg^2 + Ba Ra}{Jm Ra} & 0 \end{bmatrix} \begin{bmatrix} \int \theta \\ \theta \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{Effm La Kt Kg}{Jm Ra} \\ \frac{-Effm La Kt Kg}{Jm Ra} \end{bmatrix} V \quad (3.12)$$

$$\begin{bmatrix} \int \theta \\ \theta \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \int \theta \\ \theta \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} V$$

Maka digunakan MathScript Node dengan memasukkan parameter motor DC pada program LabVIEW:



Gambar 3.6 MathScript Node dengan Model State Space Integrator

Dengan perhitungan matriks yang didesain pada program LabVIEW maka didapatkan hasil persamaan:

Equation

$$\frac{dx}{dt} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 307,692 & -28,0193 & 0 \\ 0 & -536,264 & 28,0193 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ 49,3217 \\ -49,3217 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 1,6277 & 0 & 0 & 0 \\ 0 & 1,63 & 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u(t)$$

Gambar 3.7 Hasil Perhitungan menggunakan LabVIEW

Dari pemodelan dan hasil perhitungan yang telah dilakukan, pengontrol dirancang dengan bantuan kode program dari LabVIEW. Dimana hal ini diperlukan untuk mencapai pilihan yang tepat dari kedua parameter, Q dan R, yang akan menyeimbangkan kepentingan relatif dari *control effort*. Untuk memperbaiki desain

dan respon sistem, berbagai Q matriks dan nilai R dicoba. Misalnya, dalam kode program LabVIEW dengan nilai matriks yang sudah ditentukan.

Tabel 3.2 Nilai Parameter Q dan R

| Matriks Q | Nilai R |
|--|---------|
| [1 0 0 0 ; 0 1 0 0 ; 0 0 1 0 ; 0 0 0 1] | 2 |
| [300 0 0 0 ; 0 1400 0 0 ; 0 0 2 0 ; 0 0 0 2] | 1 |
| [1800 0 0 0 ; 0 1 0 0 ; 0 0 1 0 ; 0 0 0 1] | 10 |

Selanjutnya menentukan kontrol optimal menggunakan matriks P dengan formulasi persamaan aljabar Riccati (A.R.E) pada persamaan 2.15, maka digunakan MathScript Node dengan memasukkan parameter:

```

A=[0 0 1 0;0 0 0 1;0 307.692 -28.0193 0;0 -536.264 28.0193 0];
B=[0;0;49.3217;-49.3217];
C=[1.6277 0 0 0;0 1.63 0 0];
D=[0;0];
sys_ss=ss(A,B,C,D);
R1=[2];
R2=[1];
R3=[10];
Q1=diag([1,1,1,1])
Q2=diag([300,1400,2,2])
Q3=diag([1800,1,1,1])
[P1,Ac1,K1]=care(A,B,Q1,R1)
[P2,Ac2,K2]=care(A,B,Q2,R2)
[P3,Ac3,K3]=care(A,B,Q3,R3)

```

Berikut masing-masing nilai Q, R dan K yang ditunjukkan pada tabel 3.3

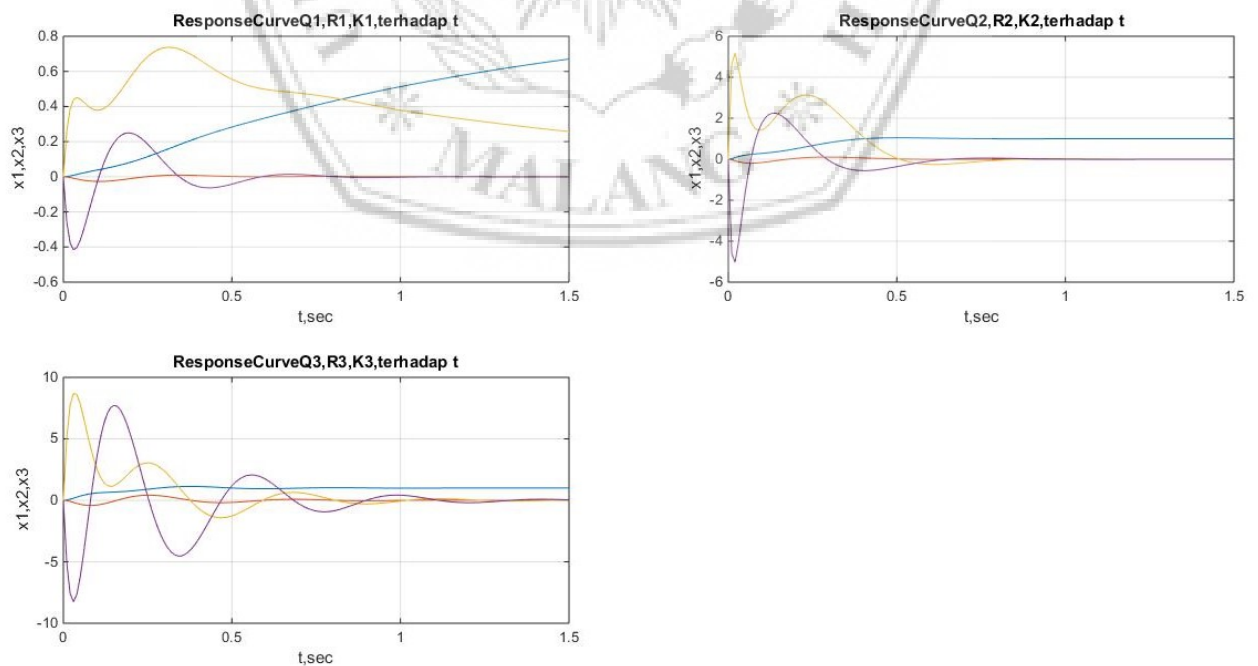
Tabel 3.3 Hasil Perhitungan Nilai Q, R dan K

| Q | R | K | | | |
|---------------------|----|---------|----------|--------|---------|
| | | K1 | K2 | K3 | K4 |
| diag [1,1,1,1] | 2 | 0.7071 | -5.3493 | 0.3927 | -0.2915 |
| diag [300,1400,2,2] | 1 | 17.3205 | -36.5534 | 2.5158 | 0.5329 |
| diag [1800,1,1,1] | 10 | 13.4164 | -0.3874 | 0.7527 | 0.2804 |

Untuk memastikan stabilitas dari sistem close loop maka digunakan rumus $Ac = (A - BK)$. Berikut nilai eigen (Ac) yang telah didapatkan:

| | |
|--------------|-----------------------|
| $A1=A-B*K1;$ | $Ac1 =$ |
| $B1=B*K1;$ | $-49.2041 + 0.0000i$ |
| $C1=C;$ | $-0.7802 + 0.0000i$ |
| $D1=C;$ | $-5.8912 + 13.1507i$ |
| | $-5.8912 - 13.1507i$ |
| $A2=A-B*K2;$ | $Ac2 =$ |
| $B2=B*K2;$ | $-96.1432 + 0.0000i$ |
| $C2=C;$ | $-16.4972 + 0.0000i$ |
| $D2=C;$ | $-6.5883 + 8.9277i$ |
| | $-6.5883 - 8.9277i$ |
| $A3=A-B*K3;$ | $Ac3 =$ |
| $B3=B*K3;$ | $-21.9708 + 13.5368i$ |
| $C3=C;$ | $-21.9708 - 13.5368i$ |
| $D3=C;$ | $-3.6869 + 14.6124i$ |
| | $-3.6869 - 14.6124i$ |

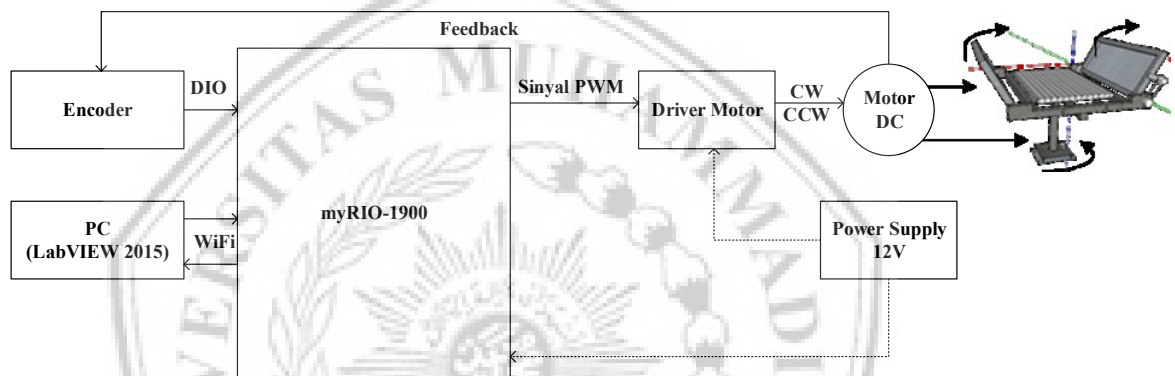
Maka didapatkan hasil tanggapan transien dari perancangan sistem ditunjukkan pada gambar 3.8



Gambar 3.8 Hasil Respon Q, R, K pada Perancangan Sistem

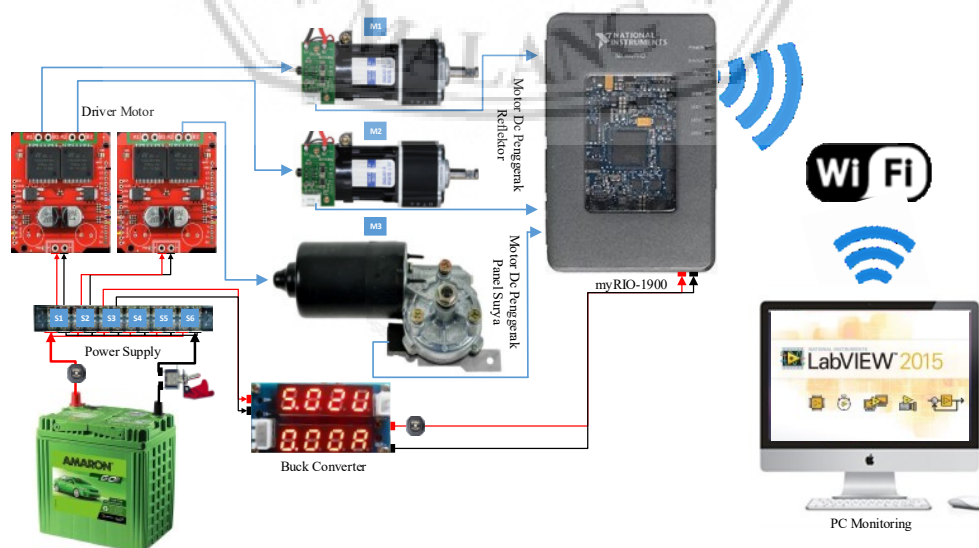
3.3 Perancangan Mekanik

Seperti yang telah dijelaskan pada bab sebelumnya, sistem *tracking* dengan reflektor didesain untuk mengoptimalkan kinerja dari panel surya dengan mengintegrasikan peralatan mekanik, elektronik dan teknologi informasi. Mekanisme ini digerakkan oleh putaran motor atau aktuator linier, yang mengontrol sistem tersebut untuk memastikan posisi optimal arah reflektor dan panel surya terhadap posisi matahari. Gambar 3.9 menunjukkan desain implementasi sistem.



Gambar 3.9 Desain Implementasi Sistem

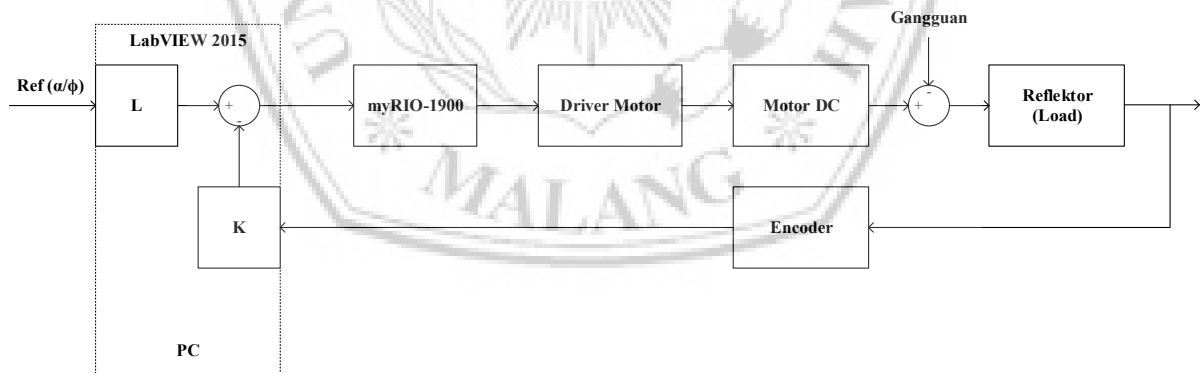
Perancangan mekanik dibuat untuk dapat memenuhi kontrol modul. Pada pembuatannya, diperlukan rangkaian pendukung hardware dan pengkabelan sistem yang ditunjukkan pada Gambar 3.10.



Gambar 3.10 Rangkaian Hardware dan Rangkaian Pengkabelan Sistem

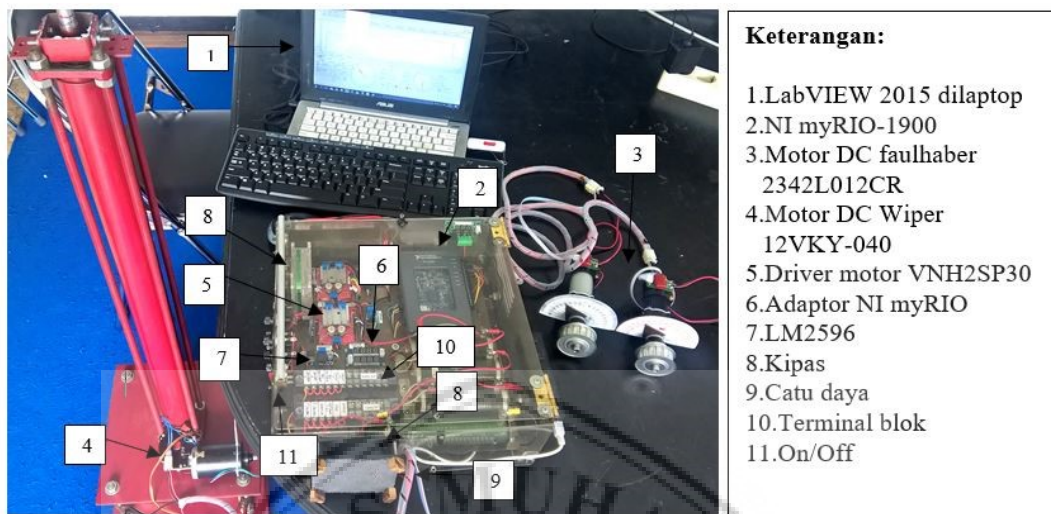
3.3.1 Perancangan Modul Kontrol

Setpoint kecepatan motor DC berasal dari data analisa dimana sistem akan memberikan perintah dari LabVIEW dengan algoritma kendali LQR yang telah dibuat, akan menghasilkan *input* untuk mengirim data tersebut ke myRIO via *WiFi*. Aktuator yang digunakan yaitu motor DC merespon *input* tersebut dengan menghasilkan perubahan posisi. Perubahan posisi tersebut diukur menggunakan sensor encoder. Sensor encoder kemudian memberikan *feedback* ke myRIO. myRIO mengolah nilai perubahan posisi tersebut menjadi kecepatan, kemudian membandingkan nilai kecepatan aktual dari *feedback* sensor encoder dengan kecepatan dari *setpoint*. Perbedaan nilai tersebut menghasilkan nilai *error* yang kemudian diolah pada program kontrol untuk menjadi *input* pada motor DC kembali. Nilai-nilai pengukuran yang didapat direkam dan dimonitoring pada komputer untuk selanjutnya dapat dianalisa dengan tujuan untuk memperbaiki respon kecepatan dari motor DC agar dapat mencapai performa yang maksimal. Maka digunakan beberapa komponen pendukung seperti yang ditunjukkan pada Gambar 3.11.



Gambar 3.11 Diagram Blok Keseluruhan Sistem

Setelah terskema dalam bentuk diagram blok selanjutnya kontroler dirancang dalam bentuk modul pengaturan motor DC yang pembuatannya menggunakan komponen-komponen yang beredar dipasaran. Selain modul, pembuatan antar muka atau sistem monitoring untuk mengamati performansi dari pergerakan motor DC digunakan software LabVIEW 2015.



Gambar 3.12 Modul Kontrol Sistem dan Monitoring Sistem Motor DC

3.3.2 Motor DC Penggerak Reflektor

Spesifikasi motor DC pada sistem ini digunakan sebagai penggerak utama pada struktur reflektor (cermin), jenis yang digunakan pada penelitian ini merupakan motor DC faulhaber yang ditambahkan dengan *gearbox* berdasarkan kecepatan *output* yang sesuai dengan spesifikasi desain struktur mekanik. Dimana diperlukan penyalan torsi yang tinggi atau percepatan yang tetap untuk kisaran kecepatan yang luas. Gambar 3.13 menunjukkan motor DC yang digunakan dalam penelitian.



Gambar 3.13 Motor DC Faulhaber

| Series 2342 ... CR | | | | | | | | |
|---|-------------------------|-----------|--------|--------|--------|--------|--------|------------------------------|
| | | 2342 S | 006 CR | 012 CR | 018 CR | 024 CR | 036 CR | 048 CR |
| 1 Nominal voltage | U_N | 6 | 12 | 18 | 24 | 36 | 48 | V |
| 2 Terminal resistance | R | 0,4 | 1,9 | 4,1 | 7,1 | 15,9 | 31,2 | Ω |
| 3 Output power | $P_{2 \max}$ | 20,5 | 17 | 18,1 | 19 | 19,4 | 17,7 | W |
| 4 Efficiency, max. | η_{\max} | 81 | 80 | 81 | 81 | 81 | 81 | % |
| 5 No-load speed | n_0 | 9 000 | 8 100 | 8 000 | 8 500 | 8 100 | 8 000 | rpm |
| 6 No-load current (with shaft \varnothing 3 mm) | I_0 | 0,17 | 0,075 | 0,048 | 0,038 | 0,024 | 0,017 | A |
| 7 Stall torque | M_H | 87,2 | 80 | 86,5 | 85,4 | 91,4 | 84,4 | mNm |
| 8 Friction torque | M_R | 0,98 | 1 | 0,99 | 0,99 | 0,99 | 0,95 | mNm |
| 9 Speed constant | k_n | 1 650 | 713 | 462 | 366 | 231 | 170 | rpm/V |
| 10 Back-EMF constant | k_E | 0,604 | 1,4 | 2,16 | 2,73 | 4,34 | 5,87 | mV/rpm |
| 11 Torque constant | k_M | 5,77 | 13,4 | 20,7 | 26,1 | 41,4 | 56,1 | mNm/A |
| 12 Current constant | k_i | 0,173 | 0,075 | 0,048 | 0,038 | 0,024 | 0,018 | A/mNm |
| 13 Slope of n-M curve | $\Delta n / \Delta M$ | 103 | 101 | 92,5 | 99,5 | 88,6 | 94,8 | rpm/mNm |
| 14 Rotor inductance | L | 13,5 | 65 | 150 | 265 | 590 | 1 050 | μH |
| 15 Mechanical time constant | τ_m | 6 | 6 | 6 | 6 | 6 | 6 | ms |
| 16 Rotor inertia | J | 5,6 | 5,7 | 6,2 | 5,8 | 6,5 | 6 | gcm ² |
| 17 Angular acceleration | α_{\max} | 160 | 140 | 140 | 150 | 140 | 140 | $\cdot 10^3 \text{ rad/s}^2$ |
| 18 Thermal resistance | $R_{th 1} / R_{th 2}$ | 3 / 15 | | | | | | KW |
| 19 Thermal time constant | τ_{w1} / τ_{w2} | 6,5 / 490 | | | | | | s |

Gambar 3.14 Spesifikasi Motor DC Faulhaber

3.3.3 Motor DC Penggerak Panel Surya

Spesifikasi motor DC panel ini di aplikasikan pada pergerakan struktur panel yang membutuhkan torsi besar dan kecepatan yang rendah, jenis yang digunakan ialah motor DC wiper yang memiliki performa tinggi yang mampu memberikan torsi hingga 100kg.cm dalam *range* kecepatan yang lebar. Terdapat dua cara yang digunakan untuk menimbulkan medan magnet motor wiper ini, yaitu tipe *wound rotor* yang menggunakan lilitan (*coil*) dan tipe *ferrite* magnet. Motor tipe *ferrite* magnet yang menggunakan magnet permanen digunakan pada motor wiper. Motor wiper terdiri dari gigi-gigi yang menyerap kecepatan keluaran dari motor. Sebuah *cam switch* tergabung didalam bagian gigi sehingga wiper akan berhenti setiap saat pada posisi yang sama. Gambar 3.15 menunjukkan motor DC wiper.



Gambar 3.15 Motor DC Wiper

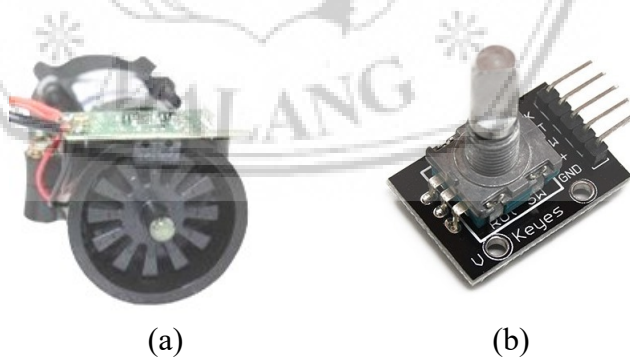
Pemilihan motor ini didasarkan pada nilai *output* kecepatan dimana motor DC wiper memiliki kecepatan output 40 RPM dengan daya yang dibutuhkan sebesar 80 W. Detail spesifikasi teknis motor DC wiper dapat dilihat pada gambar 3.16.

| | | | |
|------------------|-------|-----------------------|--------------------|
| Rating Voltage | U_n | 12V | 24V |
| Rating Power | P_n | 80W | |
| No Load Current | I_n | 2.0/2.5A | 1.5/2.0A |
| No Load Speed | n_n | 40/55 \pm 5 RPM | |
| Load Current | I_L | 10/15A | 5.0/7.5A |
| Load Speed | n_L | 30/45 \pm 5 RPM | |
| Stall Current | I | 43A _{max} | 27A _{max} |
| Stall Torque | M_A | 48NM _(min) | |
| Protection Class | | IP33 | |
| Weight | W | 2.9kg | |

Gambar 3.16 Spesifikasi Motor DC Wiper

3.3.4 Sensor Encoder

Sensor encoder yang digunakan pada penelitian ini ada 2 tipe, yaitu sensor encoder pada motor DC reflektor menggunakan tipe 2342L012CR dan *rotary* encoder pada motor DC panel menggunakan tipe KY-040 yang dihubungkan pada setiap masing-masing poros motor DC.



Gambar 3.17 (a) Sensor Encoder 2342L012CR, (b) Rotary Encoder KY-040

Sensor encoder 2342L012CR merupakan *incremental rotary* dengan resolusi sebesar $7,5^0$. Resolusi/revolusi encoder tersebut merupakan satuan perubahan posisi dapat diukur oleh encoder. Besar resolusi ini didapatkan dari besar putar encoder dibagi dengan jumlah perubahan *output* biner dalam 1 kali putar yaitu ditunjukkan pada persamaan 3.13.

$$Resolusi = \frac{\text{Sudut dalam 1 putaran}}{\text{Jumlah pulsa} \times \text{Jumlah perubahan biner per pulsa}} \quad (3.13)$$

$$Resolusi = \frac{360^0}{12 \times 4} = 7,5^0 \quad (3.14)$$

Sensor encoder *rotary* encoder KY-040 merupakan komponen digital yang dapat menghitung jumlah putaran secara presisi dengan rentang jenjang untuk rotasi penuh: 20 pulsa artinya resolusi yang didapatkan sebesar $4,5^0$ dan nilai hitungan rotasi hingga 360^0 .

3.3.5 Sistem Transmisi

Pada perancangan ini penggerak menggunakan *pulley* dan *timing belt* kemudian dibentuk katrol untuk memperkecil torsi agar didapatkan efisiensi dari sisi mekanik. Poros motor DC reflektor dihubungkan ke *pulley* menggunakan *coupling* bawaan motor. Untuk mendapatkan kecepatan yang sama, poros motor DC dengan *pulley* dihubungkan dengan *timing belt* ke poros as reflektor dengan *pulley* yang ukuran diameternya disamakan.



Gambar 3.18 Sistem Transmisi Reflektor dan Panel Surya

Dalam perancangan ini transmisi reflektor ini dapat bergerak ke arah 0 derajat sampai dengan 180 derajat. Penentuan *pulley* dilakukan dengan uji kelayakan sistem. *Pulley* yang digunakan dalam perancangan ini sebagai berikut :

Tabel 3.4 Spesifikasi Pulley

| | |
|----------------|-------|
| Diameter luar | 4 cm |
| Diameter dalam | 1 cm |
| Jarak | 15 cm |

Pada perancangan ini diameter *pulley* penggerak dan yang digerakkan adalah sama, agar bisa mendapatkan sudut yang presisi dan daya rendah. *Pulley* pada motor dan *pulley* pada reflektor dihubungkan dengan *timing belt*, sehingga dapat menggerakkan reflektor kesudut yang ditentukan.

3.3.6 Reflektor

Pada perancangan ini reflektor menggunakan cermin datar dengan bingkai sebagai penahannya. Ukuran untuk cermin yang digunakan sebagai berikut :

Tabel 3.5 Spesifikasi Cermin

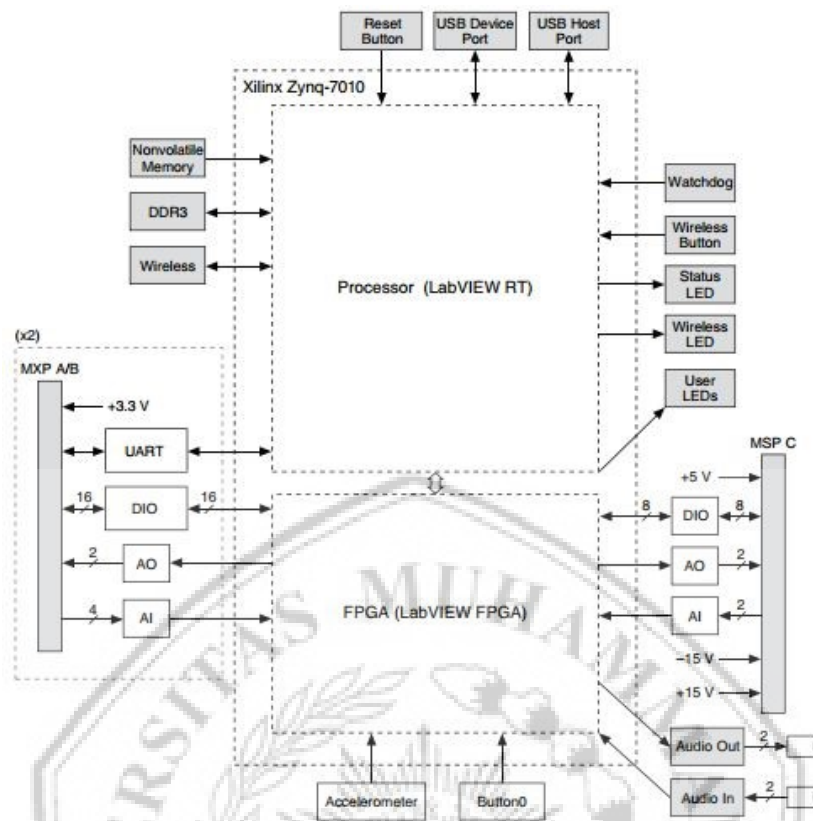
| | |
|---------|--------|
| Panjang | 100 cm |
| Lebar | 35 cm |
| Tebal | 0,3 cm |



Gambar 3.19 Cermin Datar dan Bingkai

3.3.7 NI myRIO-1900

NI myRIO-1900 berfungsi sebagai pusat pengolah data sensor, dan pengendali motor DC sesuai dengan algoritma yang diprogramkan. NI myRIO-1900 memiliki fitur yang mampu memenuhi kebutuhan sistem baik secara *hardware* elektronik maupun memori program.



Gambar 3.20 Blok Diagram NI myRIO-1900

Beberapa fitur yang dimiliki oleh NI myRIO-1900 antara lain:

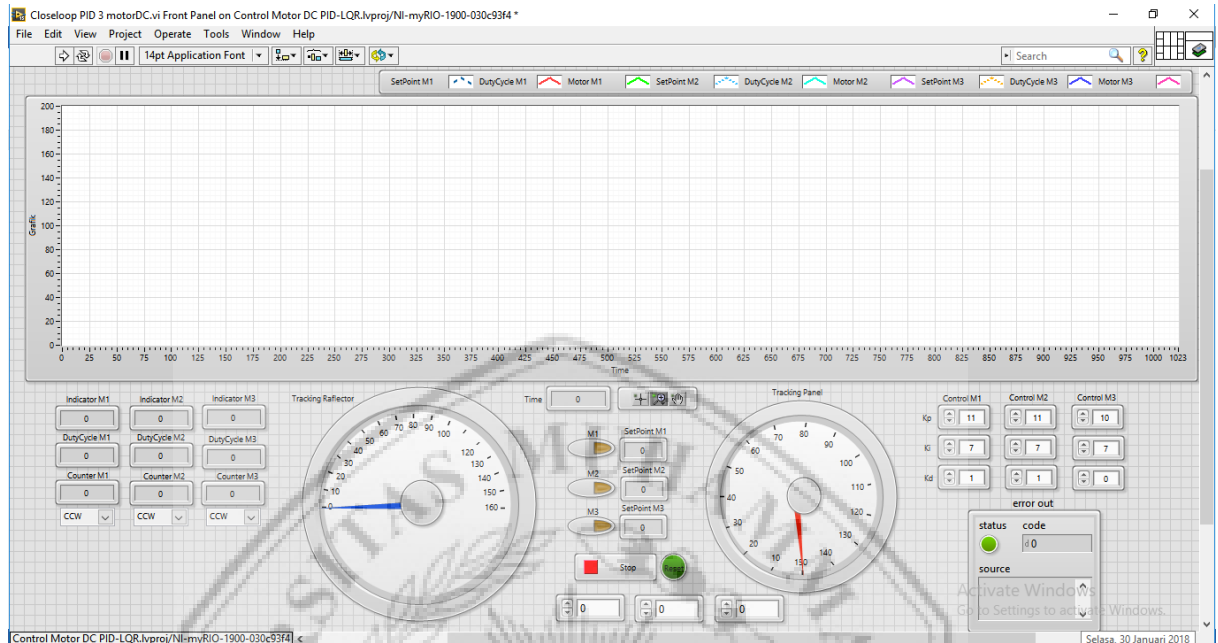
- Processor/FPGA type..... Xilinx Z-7010
- Processor speed..... 667 MHz
- Processor cores..... 2
- Memory DDR3..... 512 MB
- Memory clock frequency..... 533 MHz
- Wireless mode..... IEEE 802.11 b,g,n
- Wireless signal range..... Up to 150 m
- Security..... WPA, WPA2, WPA2-E
- Digital I/O MXP connectors..... 2 ports of 32 DIO lines
- Digital I/O MSP connectors..... 1 ports of 8 DIO lines
- Power supply voltage range..... 6-16 VDC
- Max. power consumption..... 14 W
- Typical idle power consumption..... 2.6 W



Gambar 3.21 Rancangan Sistem Tracking Dual-Axis Panel Surya

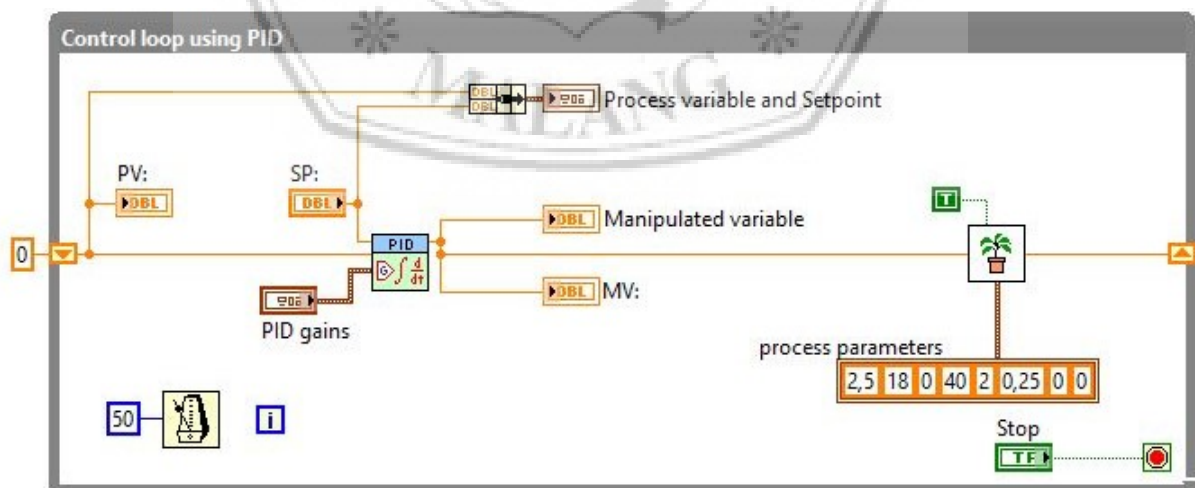
3.3.8 Perancangan Software

Pada perancangan sistem kontrol PID dan LQR ini yang perlu dilakukan adalah mengatur parameter P, I, D dan nilai referensi indeks performansi K agar tanggapan sinyal keluaran sistem sesuai masukan sedangkan Q dan R untuk variasi nilai kecepatan sebagaimana yang diinginkan sehingga mendapat nilai yang ideal, dengan melakukan perbandingan kedua kontroler tersebut. Program ini berfungsi sebagai aksi kontrol loop berdasarkan algoritma kedua kontroler, serta melakukan monitoring terhadap *plant* motor DC. Gambar 3.22 adalah rancangan *front panel* yang digunakan sebagai monitoring sistem.

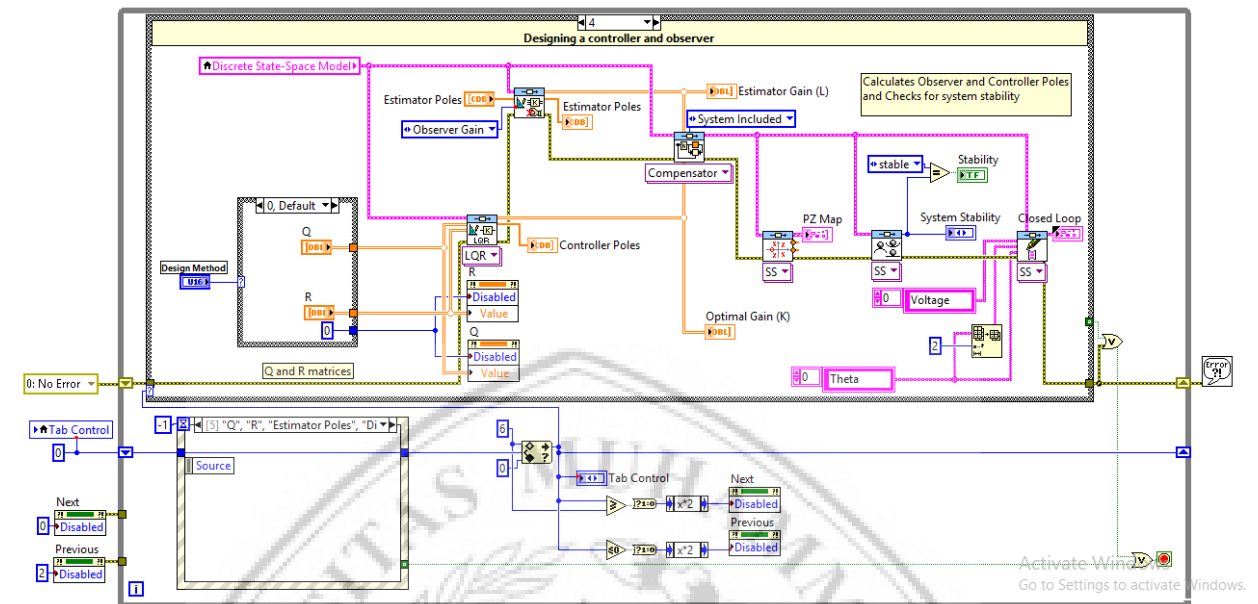


Gambar 3.22 Rancangan Front Panel pada LabVIEW

Pada tampilan *front panel* memiliki fungsi sebagai pengontrol dan mengetahui tanggapan sistem dari motor DC yang mempermudah dalam menjalankannya.

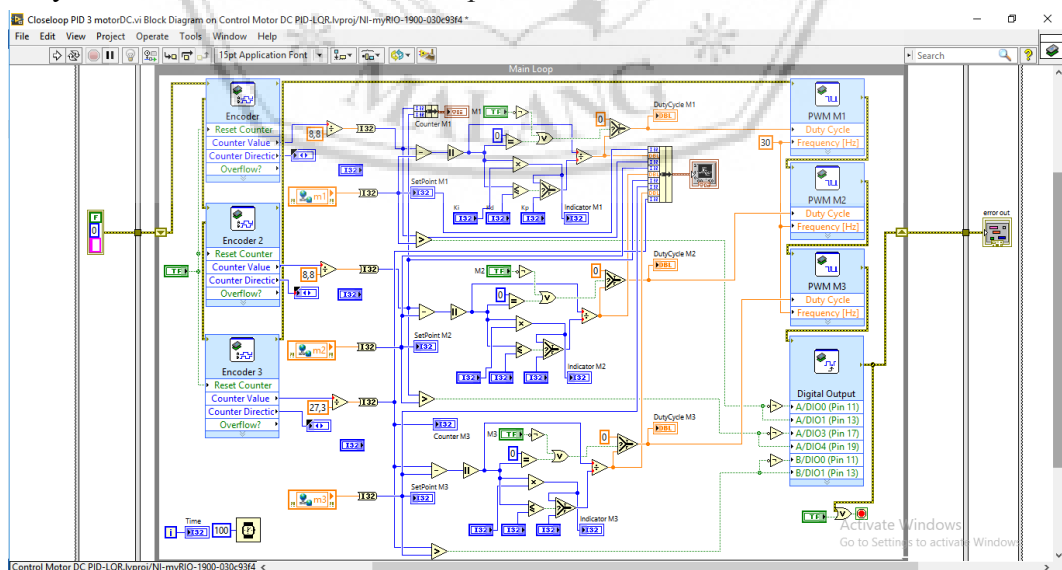


Gambar 3.23 Rancangan Blok Diagram kontroler PID pada LabVIEW



Gambar 3.24 Rancangan Blok Diagram kontroler LQR pada LabVIEW

Hasil pengujian ini dilakukan untuk membandingkan performa kontrol LQR dan PID yang bertujuan untuk mengetahui perbandingan respon transien dari motor ketika diberi kontrol LQR dan PID. Ada beberapa aspek transien yang akan dilihat dari pengujian ini seperti *rise time*, *maximum overshoot*, *settling time*, dan sinyal kontrol serta *error* antara *output* dan masukan referensi.



Gambar 3.25 Rancangan Blok Diagram Kontroler pada LabVIEW